


Hack WiFi with Crunch or Hash Cat – No Dictionaries Needed

By Hack WiFi | November 11, 2013

0 Comment

If you enjoyed this, please share it with others!

Share this with others   Follow  Share



As you've probably discovered so far, there are tons of ways we can hack WiFi passwords, be they WEP or WPA/WPA2. For network security professionals, you need to muster all the troops you can get to help you in your wireless network audits. By this I mean tools. Network security professionals need a vast range of hacking tools to assist them. The more they have available to them, the greater their changes of success. In this example, I am going to show you how to use another utility, called Crunch, to hack WiFi networks encrypted with WPA or WPA2. [Crunch](#) is an easy way to try to crack WPA passwords without

using dictionary files. Sometimes, your [WPA dictionary attacks](#) fail, and the access point you're targeting doesn't use WPS, so a [WPS attack](#) is out too. What are you left with?

Give Crunch a Try – It Can Hack WiFi Too

Crunch is not like most password hacking tools most security professionals will use. Crunch is a wordlist generator. It can calculate combinations of letters, numbers, and symbols, and then test your password hashes against all the combinations. This is a brute force attack, so it should be your last resort when dictionaries fail and WPS hacking isn't an option. There are a few caveats with using Crunch to hack WiFi keys. The first thing to keep in mind is that you'll still need to capture a WPA or WPA2 handshake. So refer back to the first half of my WPA cracking tutorial linked above. It walks you through capturing the handshake.

The command we will be using to try and hack WiFi is relatively simple. But it will take a bit of time to type out, and make sure you don't have any mistakes!

Open a terminal window and type:

```
crunch 8 12 abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789 | air-  
crack-ng -bssid 00:11:22:33:44:55 -w- hack-wifi-01.cap
```

We're basically telling Crunch to auto-generate a list of passwords with a minimum of 8 characters and a maximum of 12 characters, and a mix of lowercase and uppercase characters with numbers thrown in as well. We'll pipe the Crunch syntax and aim it at our WPA handshake capture file we sniffed in the beginning of our [other tutorial](#). I'll break down the command for you in proper fashion:

- The **8** and **12** just tell Crunch to auto-generate a brute force list with a minimum of eight characters and a max of twelve. Since WPA requires at least eight characters we can save time by not testing anything under eight. I capped the number of characters tested at 12, but you may want to do your own research on the average length of a WPA passphrase.
- What comes after is the alphabet in lowercase and then uppercase followed by the numbers zero through 9. Crunch will use this information to generate passwords of at least 8 characters and no greater than 12, all using the lowercase and uppercase letters with numbers.
- | **aircrack-ng -bssid 00:11:22:33:44:55 -w- hack-wifi-01.cap** – We will need to point Crunch to aircrack, and specify our target network's BSSID and the handshake we captured. In my original WPA hack WiFi tutorial, my target network had a BSSID of 00:11:22:33:44:55 and I had named the capture file "hack-wifi.01.cap" Obviously your target's BSSID and the name of your capture file may be different, so substitute accordingly. Know what you are doing!

Remember, knowing how to hack WiFi, actually understanding the mechanics behind it, is what separates the

good network security professionals from the keyboard jockeys.

Now you are ready to use Crunch to break the WPA key. This alternate method may crack the password because it relies on brute forcing all combinations of a password rather than specific words in a dictionary.

More Troops for the Attack – Hack WiFi Using Hash Cat

An alternative to Crunch is using [Hash Cat](#) to hack the WPA or WPA2 password. If you use HashCat, you'll need to first [convert your .cap file to a .hccap file](#). And as long as you're using the latest version Back Track or Kali Linux, you should just be able to use aircrack to convert your .cap file to a .hccap file. For instance, if the name of your capture file is "hack-wifi-01.cap", just run:



```
aircrack-ng hack-wifi-01.cap -J capture
```

Hashcat needs the .hccap file and cannot use the .cap like Crunch can. From Kali Linux, you can get to hashcat from /usr/share/oclhashcat-plus. To run Hash Cat, just type the command below from Hash Cat's file location:

```
Hashcat-plus.bin -m 2500 -a3 hack-wifi-01.hccap abcdefghijklmnopqrstuvwxyzABCDE-  
FGHIJKLMNOPQRSTUVWXYZ0123456789 pause
```

- **-m 2500** tells Hash Cat to test in WPA/WPA2 mode
- **-a3** tells Hash Cat to use brute force mode, and we need to point it at "hack-wifi-01.hccap" which is my converted capture file containing the WPA/WPA2 handshake
- As with Crunch, we can specify a character set to include in the brute force attempt.
- We should use the **pause** switch to throttle Hash Cat's cracking attempts.

So there you have it, two alternative brute force methods to hack WiFi networks, specifically encrypted with WPA or WPA2. Remember these are very time intensive attacks, but because of their nature, they are almost guaranteed to crack the password. But you have to ask, how long? Brute forcing complex WPA/WPA2 passwords could take YEARS. Or hundreds of years. Or hundreds of thousands of years. As with most WPA2 attacks, this one is in no way guaranteed to work any time soon. But, it's yet another useful tool you should keep in your IT Security toolbox.

Remember, once you’ve cracked the key, you should verify you can connect to the network by looking at the [enabling WiFi](#) tutorial for Kali Linux.

 Repost

1 ▾


If you enjoyed this, please share it with others!


Share this with others

 Follow

 Share

Google+

 Hack WiFi

 Follow

66

Category: WPA/WPA2 Attacks Tags: brute force , crack wpa , crunch , hack wifi , hash cat

infolinks